Generalised Random Forest Space Overview

Miron B. Kursa

Interdisciplinary Centre for Mathematical and Computational Modelling (ICM),
University of Warsaw,
Pawińskiego 5A, 02-106 Warsaw, Poland
M.Kursa@icm.edu.pl

Abstract. Assuming a view of the Random Forest as a special case of a nested ensemble of interchangeable modules, we construct a generalisation space allowing one to easily develop novel methods based on this algorithm. We discuss the role and required properties of modules at each level, especially in context of some already proposed RF generalisations.

1 Introduction

Random Forest (RF) is a popular, powerful ensemble machine learning method proposed by Breiman [2001]. Although the canonical version of this algorithm is known to be very versatile and perform well in numerous applications, many variants of this method have been proposed, for many different purposes: to extend the RF capabilities, to generalise over specific, non-standard data, to increase accuracy in a certain conditions, to improve the attribute importance measure produced by this method, to speed-up training or prediction, just to name a few.

This work aims to provide a conceptual framework of generalised Random Forest (GRF) methods, useful both in classification of existing RF variants and defining research opportunities in this field.

2 Generalised Random Forest

In the presented model of the generalised Random Forest, we assume a following three-layer nested ensemble structure, as shown on Figure 1.

The data is ingested into the model via numerous *pivot models*, which have to fit the form of the data and are expected to be very simple and easy to train, although are not necessary have to be neither accurate nor robust; their intended role is to become an interface between the input and internal logic of the GRF model. In other words, we assume that they handle the feature construction step of the modelling process.

Pivot models are grouped are converted into a meaningful ensemble models with the *sharpening ensembles*; this modules of this class orchestrate the generation of pivots by using the information from the decision attribute. The sharpening ensemble is expected to produce accurate models regardless of their robustness; however, they must not require external optimisation of any kind and should be computationally efficient.

The outermost layer of the GRF is the *conditioning ensemble* which builds and groups sharpening ensembles. Its role is to finally build a robust model by joining a lot of accurate models for which one does not know which are overfitted and which are not. It is trivial to show that this can be achieved even with a very small fraction of meaningful models, provided that sharpening ensembles will not be correlated; to this end the conditioning ensemble usually involves some permutation strategy.

Moreover, conditioning ensemble can be expanded to provide a generalisation of some of the additional features of Random Forest, like internal error approximation, object-object dissimilarity measure or feature importance. It is also a good place to implement parallel computing capabilities.

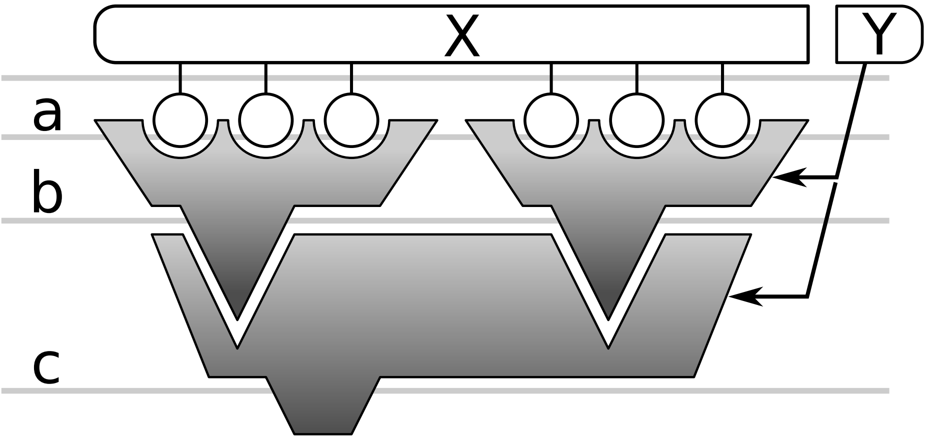

Fig. 1. Structure of a Generalised Random Forest. a) pivots, b) sharpening ensemble, c) conditioning ensemble. X and Y denote, respectively, predictor and decision part of the information system.

3 Pivot models

Pivot models are the only part of the GRF structure that has a direct contact with predictors of an information system, thus it is a place for modifying the interface with the data structure. The form of an output of a pivot classifier and the algorithm which is used to train it is naturally enforced by a form of the sharpening ensemble; most often pivot forms an embranchment in some kind of a decision tree, thus shall return a direction in which certain object should descent within the tree. As the trees are most often binary, this gives us two options, which we will later in the paper call R (for right) and L (for left).

In a standard Random Forest [Breiman, 2001], it is assumed that an information system is composed of either categorical or continuous[1] predictors. This way, we have only two possible pivot classifiers, respectively for categorical and continuous feature, in a following forms:

$$f(x \in \mathcal{C}_x; \varXi \in 2^{\mathcal{C}_x}) \rightarrow \{R, L\} := \begin{cases} R : x \in \varXi, \\ L : x \notin \varXi, \end{cases}, \tag{1}$$

where $\mathcal{C}_x$ is a set of categories of x, and

$$f(x \in \mathbb{R}; \varXi \in \mathbb{R}) \rightarrow \{R, L\} := \begin{cases} R : x \geq \varXi, \\ L : x < \varXi, \end{cases}. \tag{2}$$

In either case, x is a feature on which the pivot is executed, and $\varXi$ is the *threshold* value. Both are selected when pivot is generated within the sharpening ensemble; most often they are optimised to maximise the decision homogeneity within the subsets of objects sent left and right, usually measured by information gain or Gini index.

The other popular idea, started by Geurts et al. [2006] with their Extra-Trees method, is to generate pivots at random, obviously with constraint that a resulting criterion should not sent all objects in one direction. Such a method removes a problem of finding an appropriate homogeneity measure and performing the optimisation, which in return increases the computational efficiency and, in a number of problems, the robustness of a final model as it leads to a greater divergence among sharpening ensemble models. Naturally, this way one also removes an impact of a possible problems with the homogeneity measure, which may surface for instance in case of unbalanced sets. Still, there are problems in the probability of finding even a slightly meaningful pivot is very small, and they usually case random pivot-based algorithm to perform poorly.

And in-between solution is to use some kind of heuristic instead of a full optimisation; a simple example of this technique is to generate some number of random pivots and select the best one. Other approaches include generic soft optimisation methods like genetic algorithms, randomly reducing the search space (often the set of used predictors, ad done by the standard Random Forest) or disturbing the homogeneity measure. Rodríguez et al. [2006] proposed to employ PCA in pivot generation.

The other way of generalising the pivot models is to modify their structure to accommodate information systems going beyond simple sets of categorical and continuous predictors. Bosch et al. [2007] perform image classification with GRF with pivots calculating two vector image descriptors, summing their valus with random weights and comparing with a random threshold. Fan [2009] proposed to employ some kernel function used for SVM and build pivot by partial application of this kernel to a randomly selected object and compare the result with a threshold optimised in terms of information gain. This approach was later used for sound and temporal gene expression patterns [Cao and Fan, 2010, Fan et al., 2010].

[1] One should note that ordered categorical data, like for instance cold < warm < hot, can be treated as continuous without any loss of generality.

4 Sharpening ensemble

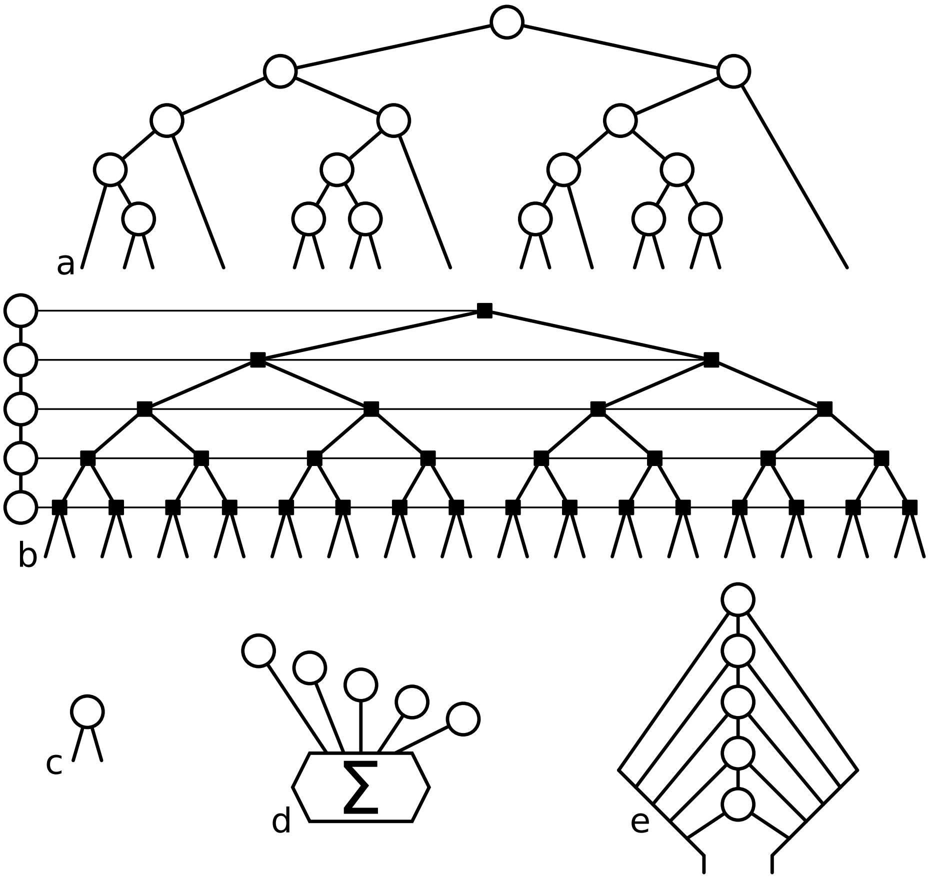

Fig. 2. Examples of sharpening ensemble structures. Embranchment pivot based: a) decision tree, b) fern, c) null ensemble; generic: d) boosting, e) decision trunk.

The selection of a sharpening module is more complex, since it gives a lot of space for different solutions and approaches. Some of a popular options, illustrated in Figure 2, are:

- **decision tree** is a sharpening module used by the Random Forest method; it uses simple embranchment pivots stacked in an iterative manner, i.e. after applying a pivot a sub-tree is build separately for each of the branches, until the sets of objects in leaves will become homogeneous or some pre-set maximum tree depth will be achieved. Training of pivots within a decision tree may be performed both randomly or via optimisation; the leaves of a tree will likely end up homogeneous either way. To this end, the output of a tree sharpening module is a direct prediction of class.

- **decision fern** [Özuysal et al., 2008, Kursa, 2014] is basically a form of a full decision tree in which each pivot module at a given depth is identical. To this end, evaluation of a given pivot is not dependent on the others and can happen in any order, which makes a fern more computationally effective than a decision tree.
  On the other hand, this makes optimisation of a fern is highly non-obvious, thus individual pivots are usually generated randomly. This way the leaves of a fern are often non-homogeneous, and thus the prediction of a fern is often encoded as a vector of class probabilities, which naturally requires appropriate adaptation of the voting scheme present in the conditioning ensemble.
- **decision trunk**, proposed by Ulfenborg et al. [2013], is composed of a flat series of pivot models similar to a fern, though requires the decision to be binary (say A or B), and employs pivot modules (*segments*) which classify into three groups, A, B meaning that it is certain that an object belongs to a respective class, and ? meaning that the decision is relayed to a next pivot classifier within the trunk. Obviously, different than in case of decision ferns, the order of trunk segments is significant because the consideration at level i comes in a context that the object was claimed undecided by segments $1,\ldots,i-1$[2]. Trunks practically cannot be implemented without optimisation of pivot models and thus they provide sharp predictions similar to decision trees.
  One should note that ternary pivot models can be realised by combining a pair of regular entrancement pivots, only modified to optimise homogeneity in one branch, respectively for class A and B; ? is then given to objects directed to second branch in both pivots and for those for which prediction of both pivots are in conflict.
- Although **boosting** [Schapire, 1990] is almost always considered as a standalone ensemble, wrapping it in another layer may lead to a better resilience to noise and mislabelled objects. Moreover, boosting has a good support for regression problems, making it a promising alternative to regression trees in context of some of their known problems in this set-up.
- there is also a degenerate option which we will call here **null ensemble**, i.e. just using a single pivot classifier. This solution can be effective in case when pivot classifiers are very good on their own and applying a more complex sharpening will only result in an increased computational load. A litmus test for such a situation is when a more complex sharpening ensemble of a dynamic complexity, such as a decision tree, is creating shallow models composed of a small number of pivots.

Moreover, sometimes the procedure of building the sharpening ensemble is modified to support the outer conditioning ensemble in de-correlation of individual sharpening ensembles. For instance, in the canonical Random Forest each

[2] This way trunks can be perceived as a discrete version of boosting.

pivot is build on a randomly generated subset of attributes; this approach is very generic and can be easily ported over other sharpeners, yet obviously a number of alternative methods can be applied as well.

5 Conditioning ensemble

As mentioned earlier, the role of the conditioning ensemble is to justify a robust prediction from a set of sharpening ensembles of an unknown reliability. To this end, this module has to either somehow assess the member models or ensure independence of members so that the noise generated by the overfitted ones will average-out during voting. The first approach is yet rarely used, mostly because having a reliable enough method of assessing robustness would in practice mean that employing the whole GRF structure is redundant.

The second approach is mostly, as in the canonical Random Forest algorithm, realised through *bootstrap aggregation* (*bagging*), also proposed by Breiman [1996], or some variation of this method. Precisely, the procedure generates a number of object sub-samples for each of the sharpening ensemble; this way they are presented with differently stressed incarnations of the training data and thus exploring a wider range of aspects and becoming less correlated. Similar approach can be applied to features or implemented as weighting instead of strict selection. The other substantial option is to simply use a very variable sharpener that will generate diverse models on its own.

6 Conclusions

By re-imagining Random Forest as a three-level nested ensemble, we propose a generic, modular framework for extending and modifying this method.

Acknowledgments. This work has been financed by the National Science Centre, grant 2011/01/N/ST6/07035.

Bibliography

Anna Bosch, Andrew Zisserman, and Xavier Munoz. Image classification using random forests and ferns. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007. ISBN 978-1-4244-1630-1. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4409066`.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, August 1996. ISSN 0885-6125, 1573-0565. doi: 10.1007/BF00058655. URL `http://link.springer.com/article/10.1007/BF00058655`.

Leo Breiman. Random forests. *Machine Learning*, 45:5–32, 2001.

Jiguo Cao and Guangzhe Fan. Signal classification using random forest with kernels. *2010 Sixth Advanced International Conference on Telecommunications*, pages 191–195, 2010. doi: 10.1109/AICT.2010.81. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5489853`.

Guangzhe Fan. Kernel-induced classification trees and random forests, 2009. URL `http://math.uwaterloo.ca/statistics-and-actuarial-science/sites/ca.statistics-and-actuarial-science/files/uploads/files/2009-06.pdf`.

Guangzhe Fan, Jiguo Cao, and Jiheng Wang. Functional data classification for temporal gene expression data with kernel-induced random forests. In *2010 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology*, volume 1, pages 1–5. IEEE, May 2010. ISBN 978-1-4244-6766-2. URL `http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5510482`.

Pierre Geurts, Damien Ernst, and Louis Wehenkel. Extremely randomized trees. *Machine Learning*, 63(1):3–42, March 2006. ISSN 0885-6125. doi: 10.1007/s10994-006-6226-1. URL `http://www.springerlink.com/index/10.1007/s10994-006-6226-1`.

Miron Bartosz Kursa. rFerns: An implementation of the random ferns method for general-purpose machine learning. *Journal of Statistical Software*, 61(10): 1–13, 2014. URL `http://www.jstatsoft.org/v61/i10/paper`.

Juan J Rodríguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–30, October 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.211. URL `http://www.ncbi.nlm.nih.gov/pubmed/16986543`.

Robert E. Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990. URL `http://link.springer.com/article/10.1007/BF00116037`.

Benjamin Ulfenborg, Karin Klinga-Levan, and Björn Olsson. Classification of tumor samples from expression data using decision trunks. *Cancer Informatics*, 12:53–66, 2013. ISSN 1176-9351. doi: 10.4137/CIN.S10356.

Mustafa Özuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *Image Processing*, 2008. doi: http://dx.doi.org/10.1109/TPAMI.2009.23.